Documents scientifiques structurés : modèles et outils d'écriture

Jérôme Cardot, Sylviane R. Schwer

Documents scientifiques structurés : modèles et outils d'écriture

Jérôme Cardot et Sylviane R. Schwer
LaLICC (UMR 8139), université Paris IV
jerome.cardot@paris4.sorbonne.fr, sylviane.schwer@paris4.sorbonne.fr

Besançon, 13-14 mai 2004

Mots clefs: Document scientifique, document structuré, document écrit, modèle orienté objet

Résumé

Les documents scientifiques écrits constituent un paradigme de document structuré : ouvrages divisés en chapitres (ou articles), eux-mêmes structurés en sections, sous-sections, comportant des figures, des bibliographies...

Une grande partie des documents scientifiques publiés sont produits avec le logiciel LaTeX, qui propose – et impose – au rédacteur de construire son document en respectant explicitement cette structuration. De la sorte le rédacteur est déchargé de tâches de mise en forme (prises en charge de façon homogène par les feuilles de style) et peut se consacrer au fond.

En outre, LaTeX n'est pas seulement un ensemble de commandes de mise en forme adapté à la production des textes scientifiques, mais aussi un langage de programmation : il permet au rédacteur de l'enrichir, de créer de nouvelles commandes, et de conserver dans le document une structuration syntaxique correspondant à la structuration sémantique vue par l'auteur, en clair, de créer de nouveaux types d'objet dans le document.

Nous présentons un modèle de document structuré, puis nous présentons LaTeX, comme outil de production de tels documents et nous le comparons à d'autres outils :

– logiciels standards de traitement de textes ;
– documents écrits à l'aide de langages à balises (XML, MathML…)

Nous insistons aussi sur le caractère libre (ouvert, connu du public) du mode de représentation interne des documents produits à l'aide de LaTeX, qui permet aussi bien d'obtenir un document LaTeX en sortie (résultat) d'un programme, que de l'utiliser en entrée d'un autre (conversion en HTML pour diffusion sur le Web, extraction d'information en traitement du langage naturel, réalisation de bases de données d'articles…)

1 Traitement et édition de textes

L'édition de documents met en jeu différentes fonctions qui étaient traditionnellement assurées par des professions différentes et que les outils modernes de publication tendent, au

moins dans certains domaines comme celui de l'édition scientifique, à concentrer sur le seul rédacteur.

Rappelons ici ces rôles, et comment les logiciels utilisés pour l'édition peuvent aider à maintenir cette distinction de rôles, ou entretenir la confusion.

1.1 Les rôles

On pouvait autrefois distinguer, dans la réalisation d'un article rédigé pour une revue, les rôles suivants :

l'auteur rédige le document ; c'est bien sur le fond, le sens de son écrit qui l'intéresse au premier chef ;

l'éditeur a la responsabilité de la revue ; il s'assure – entre autres – de l'allure de celle-ci, en définissant une présentation homogène que devront respecter les documents, et les numéros successifs de la revue ;

le typographe réalise matériellement la revue, disposant les caractères du texte de l'article suivant la présentation-type à respecter.

L'évolution des techniques de production de documents fait désormais reposer la tâche dévolue au typographe sur l'auteur du document ; l'éditeur lui transmet ses consignes (taille des caractères, canevas...). Suivant le type de logiciel qu'il utilise, l'auteur devra alors consacrer une énergie plus ou moins grande à respecter lesdites consignes.

1.2 Logiciels

Parmi les logiciels à la disposition du rédacteur, on distingue :

les éditeurs de textes qui permettent d'écrire et d'enregistrer des chaînes de caractères ; si le texte écrit doit respecter certaines contraintes (par exemple pour les textes sources de programmes) l'éditeur peut aider le rédacteur en coloriant certains éléments syntaxiques (mots clés et commandes, ponctuation, parenthésage...) ; tous les systèmes informatiques sont distribués avec au moins un éditeur de texte ; l'éditeur `emacs` (et sa variante pour les sytèmes à interface graphique `XEmacs`) sont une référence, et reconnaissent les mots-clés de la plupart des langages de programmation.

les logiciels de traitement de textes reprennent les tâches dévolues aux éditeurs de texte, en leur ajoutant des commandes de mise en forme du texte ; ils permettent donc de créer et de mettre en forme la plupart des documents destinés à l'impression : lettres, articles, livres...

les logiciels de publication assistée par ordinateur offrent de nombreuses possibilités de présentation. Leur but est de permettre la mise en page de tous documents écrits, en particulier revues et magazines : ils permettent entre autres de définir des maquettes, des zones de texte sur la page, des effets graphiques sur les caractères et les photographies.

Dans cette catégorie de logiciels, la référence des utilisateurs professionnels est `Quark XPress`. Le logiciel libre `Scribus` reprend des concepts analogues. Ces logiciels privilégent l'aspect final du document, et leur utilisation ne saurait donc s'abstraire du support physique définitif.

2 Documents structurés

Dans une approche de gestion de documents, qui pourrait être celle d'un bibliothécaire ou d'un éditeur, on s'attachera à structurer les informations qui composent le document ; la figure 1, inspirée de [Ham96] représente ainsi les éléments (dans le vocabulaire des modèles à objets, les entités) de cette structure.

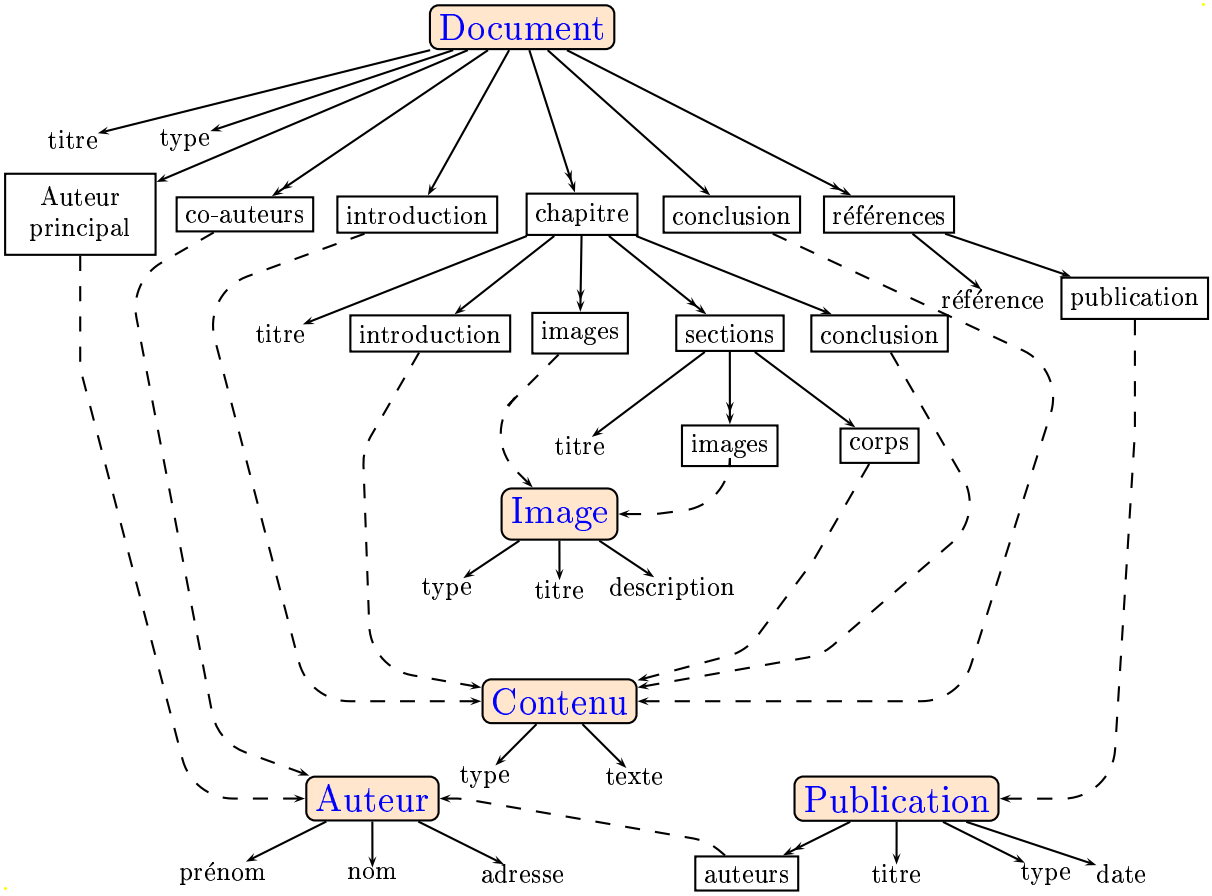

FIG. 1 – Modèle de document

Sur cette figure, les objets (document, image…) sont dotés de propriétés directes (non encadrées sur la figure) et d'éléments (encadrés sur la figure) qui suivent eux-mêmes la structure d'autres objets (ce qui est symbolisé par les pointillés). La double flèche indique que certains éléments peuvent être présents en plusieurs exemplaires dans un objet, par exemple il peut y avoir plusieurs images dans un chapitre.

Une telle structuration correspond bien à la représentation que peut avoir l'auteur du document : dans un premier temps ce qui importe pour lui, c'est l'enchaînement des différentes parties, indépendemment de leur réalisation physique (répartition du texte en pages, ou affichage à l'écran…)

Par ailleurs, le retour à la connaissance de cette structure est nécessaire pour certains traitements ultéreurs d la chaîne éditoriale, comme la réalisation des tables des matières, table des

illustrations… qui seront réalisées après la phase de mise en page (ou leurs équivalents hypertextuels qui seront réalisés lors de la phase de mise en écrans).

Si ce besoin de structuration du texte peut sembler mineur à l'auteur d'un roman (mais certainement pas à son éditeur ou au bibliothécaire), il est tout à fait perceptible à l'auteur d'un texte scientifique (toutes disciplines confondues) dont l'argumentation est soutenue par un découpage en parties, sous-parties…

Dans le domaine littéraire, le découpage d'un texte de théâtre en actes, scènes, didascalies et répliques est d'ailleurs tout-à-fait analogue.

C'est donc la structure du texte qui importe essentiellement à l'auteur, la mise en forme selon un canevas donné pouvant être vue comme le résultat d'une série d'instructions de transformations (c'est-à-dire d'un programme, au sens informatique du terme) opérant sur les données que constitue le texte structuré.

3 LaTeX

3.1 Présentation

Un document LaTeX s'écrit, comme un programme, à l'aide d'un éditeur de textes. Le document appelé *texte source* comprend, outre le texte lui-même, des instructions de structuration. Tout le document, y compris les commandes et les formules mathématiques, est écrit sous la forme d'un texte ordinaire (avec les caractères standards disponibles sur tout clavier).

Il est ensuite nécessaire de transformer ce texte source en un fichier visualisable : cette phase, appelée *compilation*, exécute les commandes du fichier source et transforme par exemple le texte

```
$$\sum_{n=0}^5 n^2$$
```

en :

$$\sum_{n=0}^{5} n^2$$

Le fichier visualisable peut être, au choix de l'utilisateur, de différents formats, par exemple au format PDF.

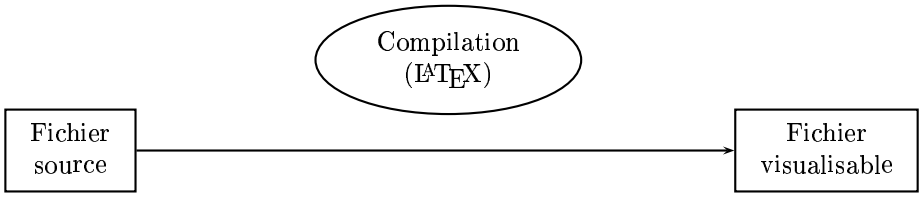

FIG. 2 – Compilation d'un document LaTeX

Le fichier sur lequel travaille l'utilisateur n'a donc pas l'allure du document produit, contrairement à d'autres traitements de textes dits WYSIWYG[1]. C'est parfois un peu moins confortable pour l'utilisateur, mais cela l'oblige à structurer consciemment son texte, et cela lui permet de travailler directement sur le document tel qu'il est connu du logiciel, ce que ne permettent pas les traitements de texte WYSIWYG (puisque ceux-ci ne montrent à l'utilisateur qu'une représentation graphique du document).

Par leur interface graphique, les logiciels de ce type (`WordPerfect`, `AbiWord`, `Word`, `OpenOffice`...) incitent l'utilisateur à mettre en forme son document d'une façon qui, dans le format interne du logiciel, oublie certains aspects de la structure.

En outre, le passaage par la compilation apporte plus de souplesse au rédacteur, qui peut faire figurer dans son code source des commentaires qui n'apparaîtront pas dans la version compilée du document, ou qui peut rédiger une équation mathématique avec des alignements différents de ceux de la version compilée : LaTeX ne prend en compte qu'un espace même si le rédacteur en a écrit beaucoup, et ne change de paragraphe que si le rédacteur laisse une ligne vide.

3.2 Séparation des rôles

Avec LaTeX, les rôles de l'auteur, de l'éditeur et du typographe sont nettement séparés : en effet l'auteur ne place dans son fichier source que les commandes de structuration du document, et la référence à une *feuille de style* fournie par l'éditeur.

Le programme LaTeX, lors de la compilation, joue alors le rôle du typographe, en mettant en forme le texte de l'auteur conformément à la feuille de styles de l'éditeur.

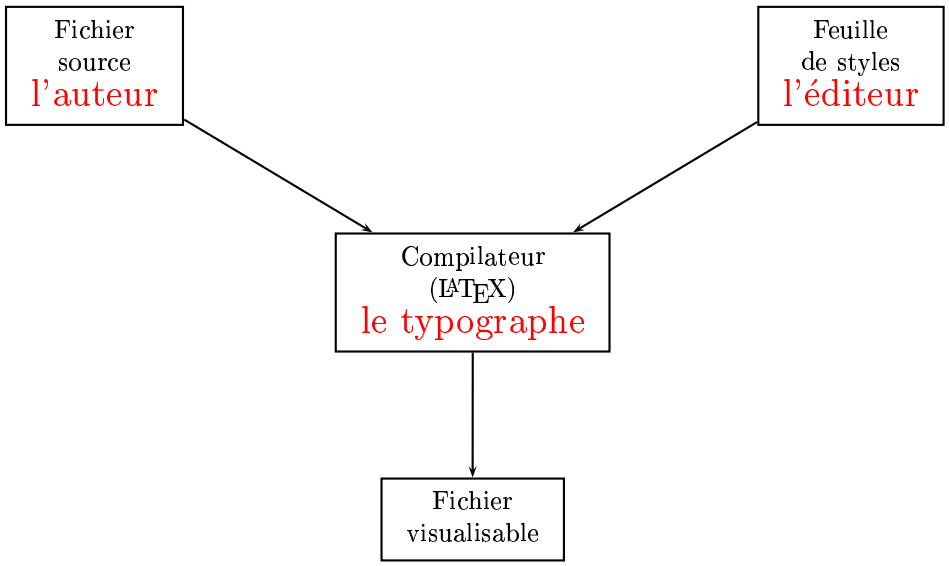

FIG. 3 – LaTeX et les rôles

[1] Acronyme de « *What You See Is What You Get* », pour lequel on utilise parfois en français « *Tel écran, tel écrit.* »

3.3 Éléments d'un fichier LaTeX

Un fichier source LaTeX aura l'allure suivante :

```
\documentclass[12pt]{article}
\usepackage{french}
\usepackage{feuille-de-style}

\title{Le titre de l'article}
\author{Le nom de l'auteur}

\begin{document}
\maketitle

\section{Une section.}
Texte...
\section{Une autre section.}
Texte...
\subsection{Avec une sous-section.}
Texte...
\subsection{Plus une autre.}
Texte...
\end{document}
```

La commande \documentclass, que l'on trouve en début de document, rattache celui-ci à l'une des classes suivantes :

- book ;
- report ;
- article ;
- letter ;
- seminar (utilisée pour réaliser des transparents).

Elle peut être suivie d'une ou plusieurs commandes \usepackage{complement}, qui peuvent définir la façon dont le texte devra être rendu (sur l'exemple, french apporte des règles typographiques spécifiques aux textes français, comme la gestion de l'espace à laisser avant les signes de ponctuation hauts ? ! ; et :. Et feuille-de-style est fournie par l'éditeur, par exemple pour régler les dimensions du papier et des marges, la taille des caractères, le type de numérotation des sections...

Quant au corps du texte, il est compris entre les commandes \begin{document} et \end{document}. Outre les caractères du texte lui-même, il comprend des commandes désignées par des mots commençant par \, qui portent sur des portions de texte délimitées par { et }. Le symbole $ est utilisé pour délimiter les expressions mathématiques dans le code source.

3.4 Enrichissement

LaTeX est conçu dès l'origine pour permettre l'édition de texte scientifiques. Nous n'entrerons pas ici dans le détail des commandes mathématiques de LaTeX, mais nous les utiliserons

pour illustrer la façon dont il permet à l'utilisateur de l'enrichir, en créant de nouvelles fonctions. En cela, LaTeX se rapproche des langages de programmation.

Par exemple, si on étudie une fonction comme :

$$F(x) : x \longmapsto \int_0^x e^{-t^2} dt$$

où l'expression écrite à droite de la flèche s'obtient par

```
\int_0^{x}e^{-t^2}dt
```

on sera amené à écrire des expressions comme $\int_0^1 e^{-t^2} dt$, $\int_0^3 e^{-t^2} dt$, $\int_0^\alpha e^{-t^2} dt$, $\int_0^\infty e^{-t^2} dt$.

Il est alors à la fois plus simple, plus structuré et plus conforme à la pensée de l'auteur de définir une fonction LaTeX supplémentaire, par

```
\newcommand{\fonc}[1]{\int_0^{#1}e^{-t^2}dt}
```

et de l'utiliser ensuite par

```
$\fonc{1}$,
$\fonc{3}$,
$\fonc{\alpha}$,
$\fonc{\infty}$.
```

Les utilisateurs de LaTeX ont développé et diffusé de nombreux jeux de commandes pour divers types de documents, que l'on peut choisir d'utiliser grâce à la commande `\usepackage`. Par exemple il existe des compléments pour l'édition de formules chimiques, de diagrammes électroniques, de dessins géométriques ou encore de partitions…

Précisons en outre que LaTeX [Lam94] constitue lui-même un enrichissement de TeX, système de traitement de textes créé par Donald Knuth [Knu84].

4 Autres solutions

D'autres solutions sont désormais utilisables, basées sur ce même principe de séparation d'un document structuré, de sa feuille de styles (donnant les régles de présentation) et du programme de construction du document à visualiser.

En particulier le langage à balises XML permet de structurer des informations textuelles, et XSLT (qui est en fait un sous-langage de XML) permet de rédiger des feuilles de styles transformant un document XML en un autre… ou en un document texte.

L'intérêt du langage XML est que la seule contrainte sur les documents est précisément d'être bien structurés[2] : les balises sont définies par l'utilisateur, ce qui permet d'envisager des sous-langages de XML pour des documents spécialisés de toutes sortes (textes, graphiques, feuilles de sondages, mathématiques…)

[2]En revanche HTML, le langage d'écriture des documents sur Internet, qui est aussi un langage à balises, n'impose pas une aussi bonne structuration du document.

Par exemple `DocBook` est un sous-langage de XML décrivant la structure de documents techniques, et l'éditeur O'Reilly l'utilise pour s'assurer de l'homogénéite de la présentation de ses ouvrages.

Conçus sur la base de XML, certains systèmes de documentations peuvent restreindre les possibilités de l'utilisateur en ne lui proposant qu'une interface graphique lui imposant de respecter la structure prévue pour le document.

5 Accès, pérénité et réutilisation des données

5.1 Logiciel libre

LaTeX est un logiciel *libre*. Cela signifie que ses auteurs l'ont diffusé, non seulement sous la forme d'un programme utilisable, mais ont aussi diffusé le code même du programme.

La pratique du logiciel libre (par opposition aux logiciels dits *propriétaires*, pour lesquels l'éditeur garde son code source comme un secret de fabrication) présente pour l'utilisateur des avantages importants, comme

- la possibilité de proposer, si c'est nécessaire, des corrections et des améliorations au programme ; ainsi les logiciels libres atteignent-ils souvent plus rapidement que les logiciels commerciaux grand public un haut niveau de qualité ;
- la possibilité de l'enrichir pour répondre à des besoins spécifiques.

Avec le code source du programme, les auteurs de logiciels libres indiquent aussi sous quelle forme sont codées les données produites et utilisées par le programme. C'est pour l'utilisateur une garantie de pérénité de ses données, car quand bien même des versions futures du programme enregistreraient des données en des formats différents, ces versions seront enrichies pour relire les formats plus anciens… ce que ne font pas toujours les éditeurs de logiciels commerciaux, contraignant ainsi les utilisateurs mettre à jour leur parc de logiciels par souci de compatibilité.

5.2 Réutilisation du format

En outre, puisque le format des données est connu, d'autres programmes peuvent soit lire, soit écrire directement des documents au format LaTeX.

LaTeX peut ainsi être intégré dans une chaîne de documentation, recevoir des données d'autres programmes (une base de données pour du publipostage, un logiciel de traitement statistique…)

De la même façon, des programmes peuvent prendre en entrée le code source d'un document LaTeX et le transformer d'une autre façon que ne le ferait le programme d'origine, par exemple sous la forme d'un document `.html` pour diffusion sur le web, ou d'un document `.rtf` pour transmettre le document à l'utilisateur d'un autre traitement de textes.

Notons que de tels convertisseurs, s'ils conservent la structure du document au mieux du format cible, doivent cependant passer par la représentation graphique des formules mathématiques, sous forme d'images qui ne pourront être modifiées facilement.

5.3 Compléments sur la structure

Parmi les fonctionnalités de LaTeX permettant de profiter au mieux de la structuration du document, on peut encore signaler :

- la gestion d'un ou plusieurs index ;
- la numérotation automatique des sections, sous-sections, figures, tableaux…
- la possibilité de faire réference à ces éléments, par leur numéro et/ou par le numéro de page où ils apparaissent ;
- BibTeX offre une gestion standardisée des références bibliographiques, mises en forme selon une feuille de styles qui peut elle aussi être personnalisée par l'éditeur.

6 Limites

LaTeX est très bien adapté à la rédaction de documents structurés, en séparant les rôles de l'auteur et de l'éditeur.

Certaines revues ont cependant d'autres exigences, de nature éditoriale, qui s'accomodent mal du texte structuré. Elles peuvent ainsi demander au rédacteur de fournir seulement le corps du texte, un autre intervenant de la chaîne éditoriale choisissant des intertitres (par exemple des phrases-clés) et les disposant « *harmonieusement* » sur la page.

Agata Jackiewicz, qui étudie les séries et les reprises d'éléments dans les textes et dans le discours, met ainsi en évidence que les intertitres peuvent être incompatibles avec la structure propre du texte (que l'auteur exprime par des marques linguistiques) soit en donnant une fausse idée de l'équilibre entre les composants du texte, soit en venant rompre des enchaînements présents dans le texte. Elle propose ainsi l'exemple suivant [Jac04] :

> **Patients avec ou sans « *aura* »**
> *Pour expliquer la douleur migraineuse, il y a* **deux théories** *concurrentes : celle de l'aura-inflammation et celle de la dysmodulation sensorielle.*
> **Pour la première**, *c'est un trouble neurologique, l'aura qui déclencherait la douleur. […] Parmi les points forts de cette idée, il y a […]*
> *Cependant, cette théorie […] se heurte à d'importantes limitations […]. Parmi les limitations majeures […] Chez […] Mais, plus encore […] De plus […]*
>
> **Une forme d'épilepsie sensorielle**
> *Ainsi, l'aura peut survenir sans qu'il y ait douleur, ou avec d'autres types de céphalées […] De fait […]*
> **Ce que je propose**, *c'est une vue radicalement différente de l'ensemble des symptômes de la migraine […]*

Dans cet exemple (*La Recherche, n°369*, pp. 36-37), la structure marquée formellement par les titres de sections entre en conflit avec la structure portée par le corps du texte.

On atteint ici la limite du domaine des documents structurés, qui est le domaine domaine privilégié de LaTeX.

Conclusion

LaTeX est très bien adapté à la rédaction de documents structurés, y compris les documents scientifiques.

Dans ce domaine – pour lequel il a été conçu – il est devenu une telle référence que bien des logiciels de calcul formel ont repris sa façon de coder les formules mathématiques, et que l'on retrouve aussi des expressions en code source LaTeX (non compilées) sur des listes de discussion mathématiques.

LaTeX est ainsi le format standard de fait pour la transcription et l'édition de textes scientifiques, son seul concurrent pouvant émerger dans un futur proche semblant être MathML, un sous-ensemble de XML dédié aux écritures mathématiques. Ce dernier, fondé sur une approche structurée proche de celle de LaTeX, se heurtera à la même limite que LaTeX : les utilisateurs du meilleur outil peuvent choisir de ne pas utiliser les fonctionnalités qui en font le meilleur outil… ce qui reste une limite acceptable.

Remerciements

Les auteurs tiennent à remercier les créateurs et contributeurs de TeX, de LaTeX et des compléments (packages) utilisés, en particulier Donald Knuth, Leslie Lamport et Timothy van Zandt, ainsi que Haider Hamza pour la première version du modèle de document et Agata Jackiewicz pour l'analyse de la structure et du manque de structure des textes.

Références

[Ham96] Haider Hamza. *An algebra for structured documents in the context of object-oriented approach.* Thèse de doctorat, INSA de Lyon, 1996.

[Jac04] Agata Jackiewicz. *Décrire et modéliser l'organisation discursive des textes.* Communicaion au séminaire LaLICC, et article soumis pour publication, 2004.

[Knu84] Donald Knuth. *The TeXbook.* Addison-Wesley, 1984.

[Lam94] Leslie Lamport. *LaTeX : a document preparation system.* Addison-Wesley, 1994.

Références Web

LaTeX Navigator Documentations et packages au LORIA.
`http://tex.loria.fr/index.html`

Foire Aux Questions LaTeX .
`http://www.grappa.univ-lille3.fr/FAQ-LaTeX/`

GUTenberg Le Groupement des Utilisateurs de TeX francophone propose TeXLive, une distribution de LaTeX.
`http://www.gutenberg.eu.org/`

TeXshop Une distribution de LaTeX pour MacIntosh.
`http://www.uoregon.edu/~koch/texshop/`

D'autres liens LaTeX proposés par les auteurs.
`http://www.lalic.paris4.sorbonne.fr/cardot/liens/LaTeX/`